# ROMEX—AN EXPERT SYSTEM TESTBED FOR TURBOMACHINERY DIAGNOSTICS


**David W. Lewis**

**Professor**

**and**

**Pradip N. Sheth**

**Associate Professor**

**Mechanical and Aerospace Engineering Department**

**Rotating Machinery and Controls (ROMAC) Laboratory**

**University of Virginia**

**Charlottesville, Virginia**


*David W. Lewis is a Professor of Mechanical and Aerospace Engineering at the University of Virginia.*

*He received B.A., B.S., and M.S. degrees from Rice University, and a Ph.D. degree from Northwestern University.*

*Dr. Lewis was formerly with IBM, Bell Helicopter, and Shell. His work now is primarily with magnetic bearings.*

*He holds seven U.S. Patents and is the joint author of more than 60 publications and technical reports.*

*Dr. Lewis is an ASME Fellow.*

*Pradip N. Sheth is an Associate Professor of Mechanical Engineering at the University of Virginia, where he has developed and taught graduate courses in manufacturing systems and is engaged in research on flexible, programmable automation, numerical control, dynamics of machinery, and CAD/CAM/CAE techniques for mechanical engineering.*

*Prior to joining the University of Virginia, Mr. Sheth held technical and management positions for about twelve years at the corporate Technology Center of Allis-Chalmers Corporation.*


## ABSTRACT

A rule based system developed for vibration oriented diagnosis of turbomachinery for fault identification and for predictive maintenance is described. The system is implemented in a PC based PROLOG environment, with the Dempster Shafer theory of belief functions utilized for evidential support of hypotheses. The direct uses of PROLOG for knowledge representation, rule interpretation, control strategy, and user interaction are described.

The vibration fault diagnosis system is considered to be one component of a comprehensive system for turbomachinery. The framework of this comprehensive system comprises hierarchical levels of generic rules (surface knowledge) and generic analytical simulation models (deep knowledge). The root level includes the surface and deep knowledge for vibration, bearings, lubricants, seals, gears and couplings, and mechanical/metallurgical aspects of fault detection. Another level comprises the generic but specific knowledge base for various categories of turbomachinery, i.e., pumps, compressors, turbines, engines. The third level includes the installation specific rules, maintenance, repair, and troubleshooting logs, and other specific usage experiences. It is shown that each component of the comprehensive system can be viewed as a distinct expert system which can be developed and utilized independently of the other subsystems while the comprehensive system is evolved over a period of time.


## INTRODUCTION

The process of diagnosing faults in turbomachinery is a multifaceted process. This process includes the use and consideration of such factors as (i) the experience knowledge base acquired from varied sources, e.g., the repair manuals, troubleshooting handbooks, experienced consultants and local plant personnel, (ii) decisions on further exploratory test/analysis of subsystems to progressively narrow down the possible causes, and (iii) analysis and interpretation of sensor data, all augmented by the local perceptions of the plant personnel responsible for maintenance and repair of the specific turbomachine. The offline troubleshooting and preventive maintenance are the traditional approaches; however, the evolution of these offline methods toward an online system which can serve as a predictive maintenance system in a significantly broader role than the traditional threshold based online alarms is a desirable goal. The application domain of both the offline and the online diagnostic systems for turbomachinery includes the power generation systems, the chemical and process applications, a variety of vehicular power plants, sewage/water treatment facilities, minerals processing applications, manufacturing equipment, and others.

Over the past four years, the conceptual framework and a research prototype of a comprehensive system for turbomachinery fault diagnosis has been developed at the Rotating Machinery and Controls (ROMAC) Laboratory of the University of Virginia. The research prototype, called ROMEX (Rotating Machinery Expert System), is continually and progressively updated to reflect the current status of the research into the evolving framework for the specification of the comprehensive system. Comments, suggestions, and validation experiences of the ROMAC industrial partners are incorporated in the evolution of the specifications for the system and in the ROMEX prototype. The industrial partners of ROMAC currently number about 50 industrial companies including turbomachinery manufacturers, pump manufacturers, and the users of turbomachinery from the utilities and the process industries. ROMEX serves as a testbed for the conceptual framework and provides a vehicle for the validation of the concepts in actual industrial settings. The current status of both, the conceptual framework and ROMEX, are described herein.

## COMPREHENSIVE SYSTEM CONCEPTUAL FRAMEWORK

An overall concept of a comprehensive diagnostics/maintenance system for turbomachinery is shown in Figure 1. It is recognized that the diagnostic procedures, including the heuristic rules and the analytical/experimental modelling techniques, can be to a large degree broadly applicable to many types of rotating machinery if the focus is on such problems as misalignment, looseness, unbalance, improper meshing of gears, cracks, and resonance. This generic system can be at the root of a hierarchical system of diagnostics subsystems. A relatively smaller set of specific (but still generic) rule base and the associated modelling schemes can be utilized for the process specific equipment such as a pump, a turbine, or a compressor. Another set of specific (but still generic) rule base and the associated modelling schemes can be directed to components such as induction motors, synchronous motors, pivoted pad fluid film bearings, specific seal configurations, and other components. Such a component oriented view of a diagnostic system has resulted in a generic diagnostic system for manufacturing equipment [1]. Similarly, the installation specific rule base and the associated database represent the relatively more specific information. Thus, the following hierarchy of genericity and, thus, a progressively expanding specificity is possible in a turbomachinery diagnostic system:

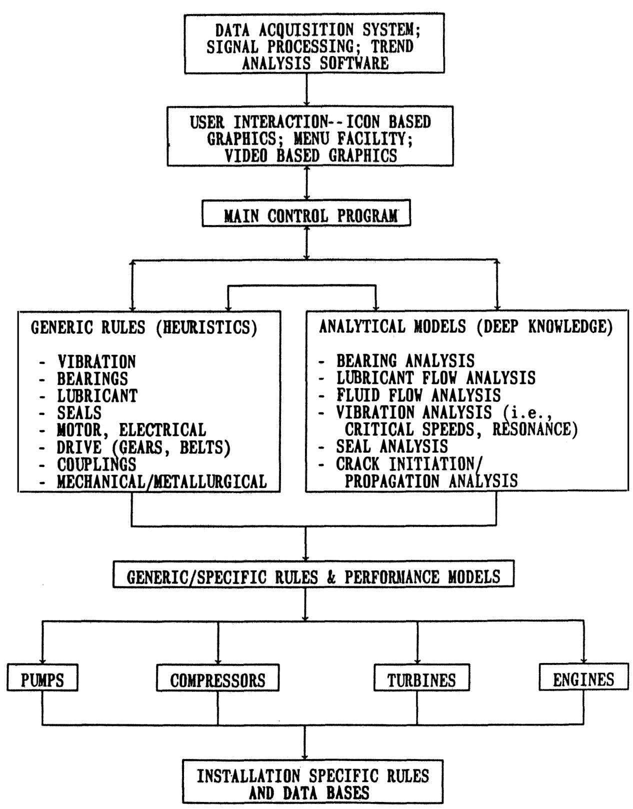


*Figure 1. Conceptual Framework of a Comprehensive Diagnostics System for Turbomachinery.*

- *Most Generic:* Rules and models applicable to a broad class of rotating machinery; in many ways, these generic rules conceptually parallel the generic algorithms for such analytical tasks as rotordynamic analysis which can be applied to a broad class of rotating machinery.

- *Generic/Specific to Process:* Rules and models, while still generic, but specific for the process application such as pumps, compressors, turbines, fans, engines, etc. For example, performance related rules and models, and vibration excitations from aero/hydraulic forces would be specifically related to the specific process but generic enough for the entire class of process.

- *Installation Specific Rules and Database:* These involve the most specific rules and database, i.e., specific to the plagued bearing, for example. The historical use, repair, and maintenance data for the specific installation would be a part of this database.

### *Symptom-Cause Relationships*

As indicated in Figure 1, the process of fault diagnosis in turbomachinery involves the interchangeable domains of symptoms and causes which manifest themselves as, for example, the vibratory behavior of the machine, the bearing performance such as the bearing temperature or power loss, lubricant data and contamination, seal behavior, and mechanical/ metallurgical observations of the components. The "symptoms" are those behavioral parameters which can be measured, analyzed, observed, or felt from the machine while the probable "causes" are inferred from these symptoms. The basic characteristics of these symptom-cause relationships are complicated by the following:

- There may be a number of symptoms about which the maintenance personnel would be uncertain because of incomplete/ uncertain information from sensors;

- Different problems may have the same symptoms and also different symptoms may result from the same problem;

- The relationships form a hierarchical structure, requiring a progressively narrowing down search procedure as more evidence of symptoms-causes is generated.

As an illustration, the symptom of high "one per REV" (i.e., synchronous with rotor speed) vibration in a radial direction would point to the generic problem of unbalance. Unbalance in an operating machine can be caused by a variety of causes including a possible loss of a part, rotor bow, or a buildup on a rotating element. Each of these causes is quite different and, once identified correctly, requires a different search strategy. The diagnostic task can, thus, be divided into two major steps. The first step involves the diagnosis of a generic problem and the second step is the refinement and the progressive narrowing down from the preliminary diagnosis of the first step. This hierarchical structure for the first two levels is illustrated in Figure 2 for vibration based diagnostics of compressors.

### *Uncertainties in Data and Inexact Rules*

The diagnostic system must be able to handle uncertainties in the data along with the varying degree of beliefs in the various cause-symptom relationships established from a number of sources. Further, the assignment of higher or lower probabilities to specific problem causes, as appropriate, from the specific maintenance history of the machine under consideration is also a necessary requirement for the system. A variety of methods for handling uncertainties in diagnostic systems are available. The subjective Bayesian approach [2], the method of

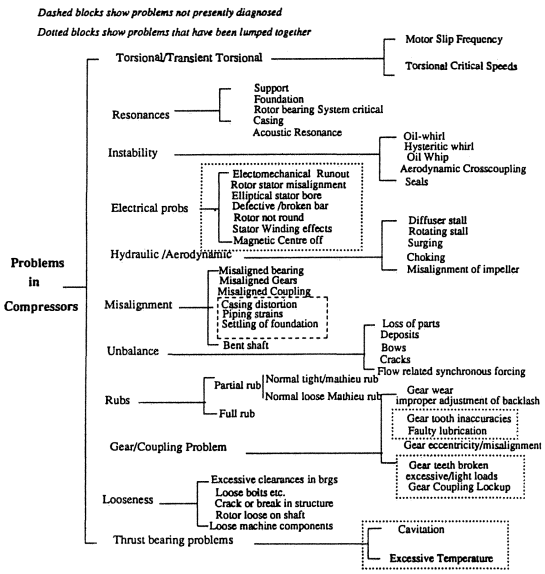


*Figure 2. Hierarchical Structure of the Diagnostic System.*

certainty factors [3], the fuzzy logic possibility theory [4], and the Dempster Shafer (DS) theory of belief functions [5] are some of the methods employed in diagnostics systems to quantify uncertainties. Of these methods, the DS theory of belief functions is particularly well suited to the turbomachinery fault diagnosis process because the progressive narrowing down of possible causes from evidentiary support is a fundamental task of the diagnostic process. For the comprehensive fault diagnostic system, the DS approach was selected, with the computational scheme proposed by Gordon and Shortliffe [6] employed to implement the DS scheme in ROMEX. The key benefits of the DS scheme include:

- The DS scheme allows for managing uncertainty in a hierarchical decision space.

- The DS method allows inexact reasoning at whatever level of abstraction that is appropriate for the evidence that has been gathered at a particular stage in the diagnostic process.

- The DS model provides the ability to distinguish between lack of belief and equal belief.

The DS method asserts that the beliefs resulting from different evidences can be combined together only if the bodies of evidence are conceptually independent. This is a key assertion for the use of the DS theory and it is necessary to exercise cautionary judgment in utilizing the DS theory when quantifying the beliefs.

*Data Acquisition*

Besides visual observations or the feeling of unusual noise or other subjective parameters, quantitative sensory data are usually available for the diagnostic process. For vibratory performance alone, a variety of probes would produce time histories and, in conjunction with an FFT analyzer system, would produce results in the frequency domain. Traditionally, various forms of data representation have been utilized for the diagnostics process, e.g.:

- orbit plots
- vibration spectrum
- time histories
- Bodé plots
- cascade plots
- polar plots, etc.

Each method of data representation is appropriate for one or more diagnostic search procedures. The diagnostic system can rely on the user to interpret the various data representations to submit the data required for the search procedure in response to a user query procedure. The interpretation of the data representations, which are performed off line, involves both one-to-one quantitative interpretation of sensory information and also a subjective assessment of the various spectra to assign the relative importance to the observed peaks and the rates of change in the various responses.

One alternative to the user interpretation of the sensory data is to incorporate a data acquisition and interpretation system which can directly interact with the fault diagnostic system and which can also interact and control the data acquisition process. A schematic representation is shown in Figure 3 of such an integrated approach for a vibration based diagnostic system. Several of the subsystems required for this integrated approach are "off the shelf", e.g., the spectrum analyzer and the IEEE-488 interface. The signal processing module, however, offers significant opportunities for innovative approaches to fault diagnostics. A neural net oriented method for pattern recognition, for example, can directly integrate the production rules of the fault diagnostic system, thus combining the diagnostic system with the sensory data analysis in a single module. This would permit the implementation of some online capabilities to effect selected corrective actions resulting from the diagnostic process.

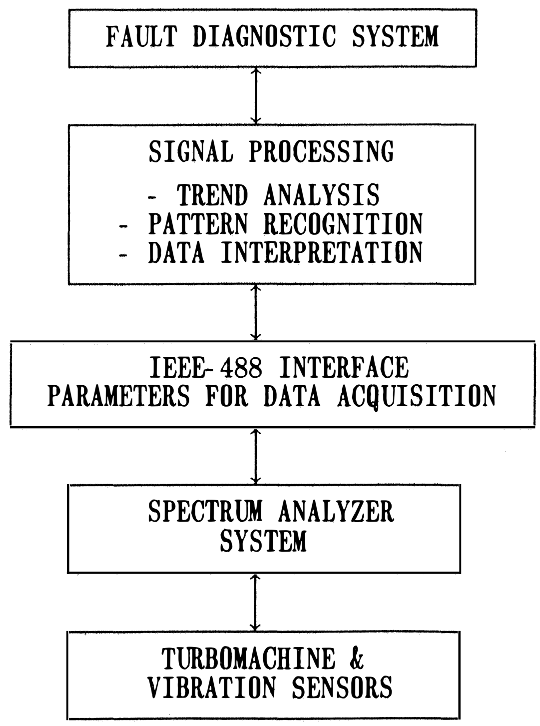


*Figure 3. Integration of Vibration Data Processing with Fault Diagnostic System.*

The facility for the user of the diagnostic system to define the specific layout and the input-output characteristics of the various sensors for the turbomachine of interest is a necessary ingredient for creating a machine data file. The available sensors would necessarily dictate the course of the user query and that of the diagnostic search process.

*Deep and Surface Models*

The current generation diagnostic system is typically a collection of "pattern action" rules which is intended to mimic the problem-solving heuristics of the expert. As discussed previously, the hierarchical search structure and the mechanisms of evidentiary support for progressively firming up the degree of beliefs in various hypotheses provide a reasonable initial prototype for the turbomachinery fault diagnostic system. This may be characterized as a surface (or shallow) system. On the other end of the spectrum, a large number of algorithmic procedures are available—primarily in FORTRAN, for rotor/bearing system dynamic analyses, stability analyses, bearing analyses, flow analyses, and fluid/structure interactions, among others. These involve a variety of modelling, analytical, and experimental analysis techniques including the finite element techniques, modal analysis, and numerical methods for the time and frequency domain solutions. ROMAC, for example, has developed a library of over 80 FORTRAN programs for turbomachinery analysis, which has been tested and validated over the past 15 years. These procedures, based on "first principles," may be designated deep models although there is no general definition yet on the form and the content of the deep models. Other possible types of deep models include: functional model [7] describing how a specific turbomachine works, detailed causal networks, and collection of rules of the form: if (symptom) then (cause) with (recommended action) which will produce (predicted response). This evolution of rules to include the predictive ability in the diagnostic system resulting from one or more recommended actions is one of several potential uses of the deep knowledge models. Within the framework of the comprehensive diagnostic system, the deep knowledge models are envisioned to complement the surface models in the following manner:

• Based on the design parameters, i.e., the structural, the mechanical, and the dynamic characteristics of the components of the turbomachine as installed at a specific site, the algorithmic procedures (including the analytical and the experimental techniques) can be utilized to establish a reference file of vibration and performance parameters. A tuned model of the turbomachine is then available to test the degree of beliefs in various hypotheses, in effect complementing the sensory data for the diagnostic process. Further, changes in the design parameters of the components over a period of time, if measurable, can be incorporated in the site model of the turbomachine. This concept of *model reference adaptive diagnostic system* is currently being evaluated as a part of the overall comprehensive system.

• The deep models can be utilized to create additional, or complementary, rules to the rules identified from expert knowledge. This idea of "compiled" deep knowledge has been utilized in a medical diagnostic system MDX [8]. In essence, the deep models are utilized at the knowledge acquisition stage to complement and perhaps validate the production rules acquired from experts.

• The deep models, as discussed, can be utilized as predictive tools for selecting and recommending appropriate corrective actions resulting from the diagnostic process.

*Organization and Growth of Production Rule Base*

As the comprehensive diagnostic system grows, the issues of discrepancies, ambiguities, redundancies, and completeness among the rules become critical. Also, a diagnostic system can never anticipate all of the potential symptom-cause relationships at the development stage. An appropriate mechanism for modifying the rules already contained in the rule base, adding to the rule base, and for changing the quantitative belief functions for various hypotheses must be provided to facilitate the growth and "learning from usage" of the system. Rule checking procedures and programs have been developed for a medical system [9]. Another approach [10] covers additional problems in knowledge-base checking by considering unreachable and deadend clauses and circular rule chains. A decision table based approach is utilized in [11] to develop a general purpose Expert System Checker written in Pascal. ROMEX currently has about 80 production rules and the knowledge base is expanding. The use of an elementary rule checker in ROMEX is currently being tested.

Closely associated with the need for a rule checker to permit a cohesive learning growth of the diagnostic system is the need for a rule base editor. Such an editor would allow the user of the diagnostic system, perhaps the plant personnel, to (i) review the existing rule base, and (ii) add to the rule base with simple, English like inputs. The editor, in turn, would utilize the rule checker to ensure a cohesive growth of the rule base. Such an editor is currently being tested [12] within ROMEX.

*Knowledge Acquisition and Validation*

This is, of course, the most critical and probably the most difficult of all of the components which constitute the diagnostic system. The intent of the diagnostic system is to mimic the thought processes of an expert to arrive at conclusions regarding the probable causes (faults) of the observed and the measured symptoms. An obvious method for creating the knowledge base would be to work through a number of case histories of turbomachinery faults with one or more experts, and hope that the experts are sufficiently prolific and the interviewer inquisitive enough to develop a probing description of the conscious and the sub-conscious thought processes involved in diagnosing a fault. It was realized, however, that the experts utilized for ROMEX were much more comfortable criticizing the rules and suggesting changes/new rules rather than starting from scratch and discussing how they go about diagnosing problems. The method utilized for knowledge acquisition and validation followed the following steps:

*Step 1:* A collection of rules for the initial knowledge base was developed from a variety of sources, including:

• Sawyer's Turbomachinery Maintenance Handbook (SOHRE Charts);

• Case studies of turbomachinery diagnostics and problem solutions from
 · ROMAC Conference Proceedings
 · Texas A&M Turbomachinery Symposium Proceedings
 · selected EPRI reports
 · interviews with the University of Virginia ROMAC faculty who are actively engaged in industrial consulting dealing with turbomachinery problems
 · Selected journal articles and books on turbomachinery maintenance

*Step 2:* The compiled knowledge base of Step 1, reflected in production rules with appropriate belief functions, was incorporated in a diagnostic system.

*Step 3:* Selected case histories made available by a ROMAC industrial partner were diagnosed using the system. Initial results were encouraging; however, this is a continuing process for the refinement and the growth of the knowledge base. The case study based approach for the validation and refinement of the knowledge base has been successfully utilized before. For example, [13] reports the use of eight actual aircraft accident cases for the confirmation and refinement of a real-time fault diagnosis expert system for aircraft applications.

The knowledge base resulting from the above steps is described in [14]. The current knowledge base is concentrated on the vibration based diagnostic process. This task, in particular, is a continuing and iterative task in nature and the evolving comprehensive system framework is expected to be significantly shaped by the progress of the knowledge acquisition and validation task.

*User/System Interaction*

There are at least three aspects of the user/system interaction:

• The input and output dialogue for defining a specific machine, its associated sensors, its repair and maintenance history, and other similar data. One possible appropriate method is to interface with one or more popular database managers such as, for example, DBASE III. The advantage of this method is that many plant personnel already utilize such systems for capturing the repair/maintenance history. For the mechanical and the structural design parameters, interface with CAD system databases would also be desirable.

• For the user query to define symptom-cause relationships, the user must have the option of asking WHY? to a specific query or to a line of reasoning. Further, the use of still photographs of the components, photographs or CAD drawings of the electrical, hydraulic, piping, or other schematics should be utilized during the user query. The SA·VANT user interface system developed for the EXACT (Expert Advisor for Combustion Turbines) system is an example of the use of a video based graphics system in a diagnostic system [15, 16].

• The user of the diagnostic system should have the option of reviewing the knowledge base by categories such as component faults, causes, symptoms, etc. The rule editor, described above, plays a vital role in this capability [12].

## ROMEX PROTOTYPE

The current ROMEX prototype [14] is directed at vibration diagnostics and contains about eighty (80) production rules. The current rulebase contains the hierarchical structure for the following problem categories (Figure 2):

· unbalance
· mechanical looseness
· misalignment
· gear Problems
· aerodynamic problems
· coupling problems
· thrust bearing problems
· subharmonic resonances
· harmonic resonances
· some electrical problems
· instability problems

The overall structure of the current diagnostic process is shown in Figure 4.

A variety of expert system shells and aids are available commercially for quick prototyping efforts. A key consideration in the development of the research prototype has been the need for flexibility. A PC-based prolog compiler, available commercially, provided the most suitable vehicle for the prototype development. Among the advantages of prolog for the research prototype, the following are particularly prominent:

• Prolog provides a strong capability for pattern matching;

• Backward chaining inference engine is already built in the prolog structure. The diagnostic system relies heavily on the backward chaining process. Also, other types of inference schemes can be relatively easily incorporated by utilizing the prolog facilities;

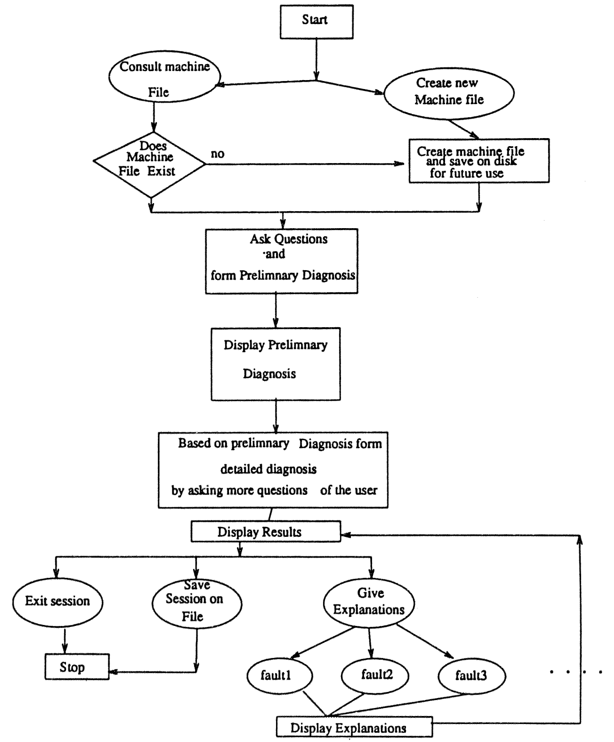


*Figure 4. General System Architecture.*

• With prolog, the capabilities of the diagnostic system can be relatively more easily expanded and modified when compared to the use of a system development tool. Note that the framework for a comprehensive diagnostic system will continue to evolve and ROMEX will need to incorporate the necessary directions defined for the framework;

• Prolog also provides a relatively easy transportability of the system for testing at various industrial partners of ROMAC.

A metainterpreter approach [17] was utilized in the prolog language to implement the following:

· mechanism for specifying certainties in rules and data;
· mechanism for computing certainties of conclusion given the certainties of the rules and the premises;
· mechanism for providing explanations.

The following is a brief description of the implementation issues.

*Knowledge Representation*

The rules and the facts are represented in ROMEX as prolog clauses. Each fact is represented as either an <object value> or <object attribute value> pair. For example: bearing (ball, inboard)—here bearing is the object, inboard is the attribute, and ball is the value of the object. The fact bearing (inboard, ball) means that bearing located on the inboard side of the compressor is of the type ballbearing. The knowledge of the fact is represented as a prolog clause "fclause/4." The first argument identifies the fact. The second argument of fclause states whether the fact is true or false or unknown (here "unknown" signifies that the user has been queried about the fact and he knows nothing whatsoever about the fact). The third argument to the fclause/4 gives the uncertainty in whether the fact is true or false. The fourth argument in fclause/4 gives the list of rules that were used to arrive at the fact. If it is a null list, it implies that the fact was arrived at by querying the user.

An example of how a fact is represented is:

fclause(bearing(ball,inboard),true,1,[ ]):

This clause states that the inboard bearing is a ball bearing with the certainty 1, and this fact was established by querying the user. The rules are represented in the following manner:

check__clause(SupA,A,B,C,D):-

B = X,Y..Z .

SupA is the super category A belongs to. A is the problem name/ category. B is a collection of premises which need to be true for A to be true. X, Y and Z represent individual premises. Each premise could either be a:

- fact
- negation of a fact
- conjunction of a fact and premise
- disjunction of a fact and premise

Additionally, each premise could also be a prolog clause. Thus, although the rule language is structured, full functionality of prolog is available to the user.

C denotes the certainty associated with the rule. D denotes the rule number. Thus, the following rule number 15 in English.

• *If* the predominant frequency is one times the running frequency and the amplitude is radial

• Then the initial problem is unbalance with certainty 0.5. would be expressed as:

check__clause(iprob,unbalance,B,0.5):-

B = pred__frequency(1),

pred×ampli__dir(radial).

*Note* that the rule itself does not specify how the uncertainty is to be computed, nor does it say how and when the facts have to be queried from the user. These tasks are accomplished by the rule interpreter.

*Rule Interpreter*

The backward chaining interpreter is the primary inference mechanism in ROMEX. The prolog predicate *solve* is the basic mechanism for implementing this interpreter. It functions roughly as follows:

• When given a goal, *solve* first checks whether it is already a fact in the memory. If the goal is found to be a fact in the memory, then *solve* checks if the certainty associated with the fact is greater than 0.1 (minimum certainty threshold), if so, then the goal is found to be true with the associated certainty, else *solve* fails.

• If the goal is not a fact in the memory, *solve* checks whether there are rules which have the given goal as the consequent. If so, then it collects the premises of the rule and makes them its new goal. *Solve* succeeds if all premises are proved to be true and the combined certainty of the rules and each of the premises exceeds the threshold. Otherwise, *solve* fails.

• If the first two conditions are not true, then *solve* checks whether the goal can be interpreted from the available facts. If so, it tries to interpret the value of the attribute to emulate "common sense" reasoning.

• If the first three conditions do not apply, then *solve* checks whether the question can be asked of the user inquiring about the truth/falsity of the fact. If so then the question is asked and users response is saved in the database. The user's response decides also whether the goal is true or false. If the goal is proved to be false, *solve* fails.

*Control Strategy*

The rule interpreter, as previously described, accomplishes the following functions. When given a goal, it determines whether the goal is true, and the belief associated with the goal. There are a few other tasks which it also accomplishes. These are:

• Formation of hypothesis.

• Combining the uncertainty in the facts generated by more than one rule.

• Displaying the results and providing explanations.

The control strategy consists of:

0) Make *Problem* as the top level node.

1) Form a hypothesis set comprising all of the subcategories of the current top level node.

2) For each hypothesis in the hypothesis set, do the following steps:

• Use *solve* to prove the negation of the hypothesis. If *solve* returns the belief in the negation of the hypothesis as 1, then remove the hypothesis from the hypothesis set and stop, else continue.

• Use *solve* to prove the hypothesis. Repeat the process until all the rules relevant to the hypothesis are utilized. Collect the beliefs in the hypothesis generated by different rules and combine them to form a composite belief (step 1 of the DS approach).

3) The DS scheme is utilized (steps 1 and 2) over the current hypothesis set, and beliefs for and against each hypothesis are combined. The hypotheses are ranked in order of beliefs associated with them. The hypothesis with belief less than 0.1 are removed from the hypothesis set, thus pruning the search space.

4) Investigate each hypothesis in the current hypothesis set by making it the top level node and going through steps 0-3.

5) Finally, all the beliefs in the various hypotheses that had been investigated are combined, and the results are saved in the database.

6) The results are displayed and, if desired, explanations are provided.

*ROMEX System Conceptualization*

The ROMEX rotating machinery diagnostic system is divided into three stages, preprocessing, processing and postprocessing. The primary function of the preprocessing stage is to gather data from the input sources and map this data into facts. The preprocessor also permits viewing and editing of the knowledge base. The processor performs the actual diagnosis. The results from this diagnosis are then passed to the postprocessor. The postprocessor presents and provides explanations of the results. The postprocessor also allows statistical analyses to be performed on the results.

*Preprocessor*

In the diagnosis of rotating machine defects, the expert uses data from a number of sources. The primary source of information is gained through vibration measurements. The expert also looks at the machine's maintenance history, the results of analytical analyses, and information gathered through machine inspection. The available data for a particular machine varies, as does the manner in which an expert uses the information. The one generality that can be made about rotating machine diagnostics is that the expert usually makes a quick, preliminary diagnosis about the machine defect. The expert then confirms or refines the diagnosis using available information and test methods.

In some cases, the machine defect may be obvious from a quick inspection of the machine, and vibration analysis is used for verification. In other cases, visual inspections and maintenance histories do not provide enough information to make a preliminary diagnosis. The expert must then rely on vibration analysis techniques to arrive at a preliminary diagnosis. The expert can use analytical or physical test methods, such as the Bump Test, to verify and refine the preliminary diagnosis.

The preprocessor has been constructed to enable the use of data from each of these sources: the machine's maintenance history, the results of analytical analyses, and vibration data. The expert system user can access machine maintenance history information, and run software simulations in the expert system environment. There are two methods for incorporating vibration data— automatically through an interface to an online machinery monitoring system, and interactively during consultation of the expert system.

A schematic layout of the preprocessor is shown in Figure 5. The preprocessor is menu driven and has six options in the top menu. Three of the options: past results, numerical analysis and online information, are used for data preparation. These options allow the expert system to access the same information that the expert uses for performing the diagnosis. The past results option contains the maintenance history of the machine. The numerical analysis option enables the user to perform an analysis on a numerical model of the machine. The types of analyses that are important to rotating machine diagnostics include critical speed, forced response and unbalanced response analyses. Many rotating machines have constant operating parameters. In this case, the numerical analysis programs only need to be executed once. The results from these analyses are stored on disk, and can be consulted by the expert system.

The knowledge gained by incorporating numerical analyses into the expert system is known as deep knowledge. Deep knowledge is defined as an explicit representation of the underlying physical principles; whereas, shallow knowledge is the term given to expert systems based on heuristics. Shallow knowledge systems are more common in diagnostic systems, but in some cases, they may not produce results with high certainty. Deep knowledge reasoning can pinpoint system dynamics characteristics very well, but some tasks that are easily accomplished using shallow knowledge representation are difficult to implement in deep knowledge reasoning. Due to the limitations of each reasoning method, a hybrid system approach is currently being explored. This approach couples shallow and deep knowledge representations and takes advantage of the respective strengths of both knowledge representations. Shallow knowledge based reasoning can be used to perform a preliminary diagnosis. In cases where shallow knowledge based reasoning cannot arrive at a diagnosis, deep knowledge based reasoning is applied.

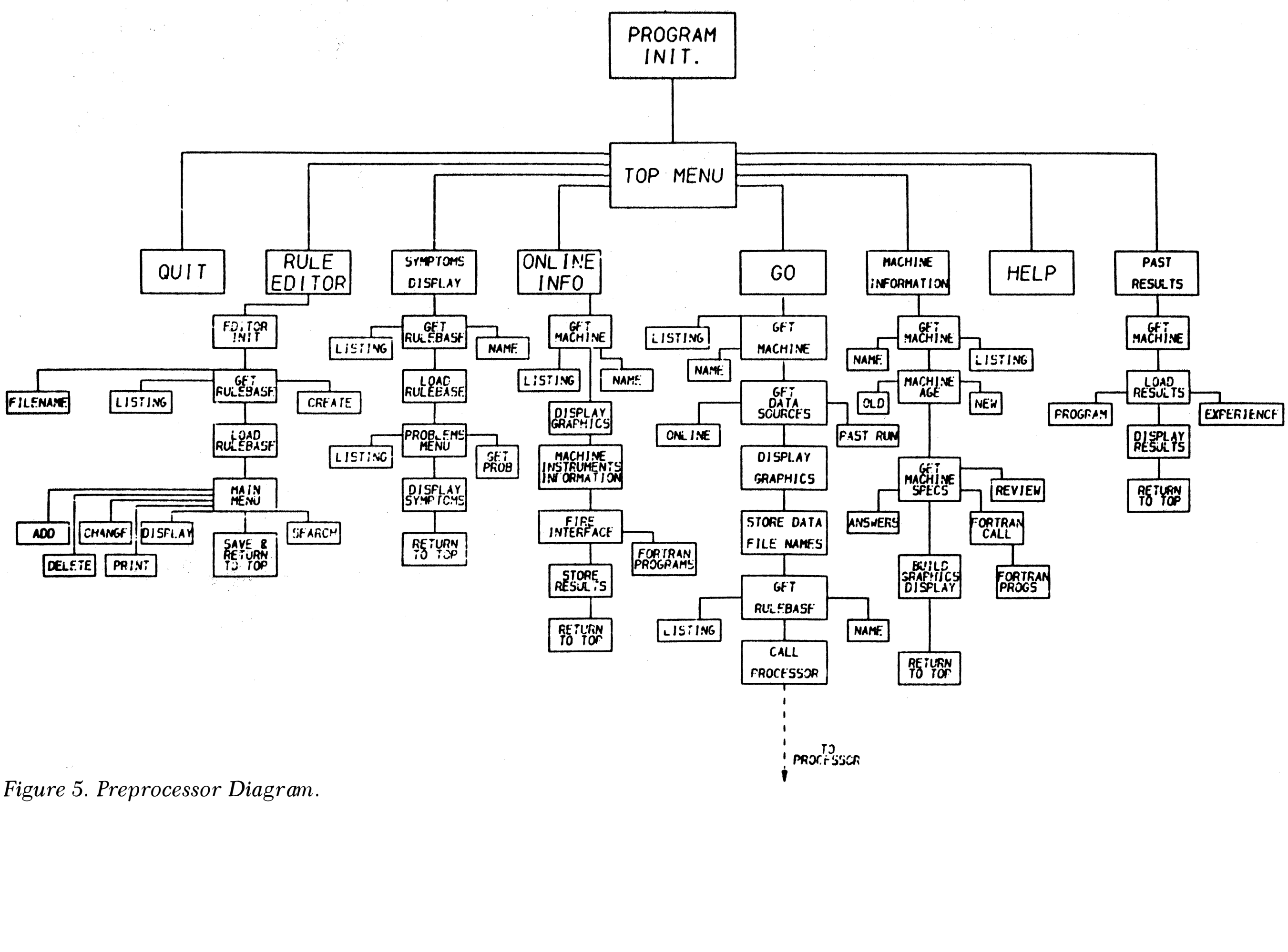


*Figure 5. Preprocessor Diagram.*

For rotating machine diagnostics, it was determined that shallow knowledge based reasoning can diagnose most problems sufficiently. In the cases where shallow knowledge representation falters, the incorporation of deep knowledge methods can add information necessary to arrive at a diagnosis. For example, operating a machine near one of its resonance frequencies is a serious problem which can cause machine failure. Since resonant frequencies of every machine are different, the rulebased expert system cannot determine if a multiple of the running speed is near one of the resonant frequencies although a resonance problem may be suspected. The resonant frequencies can be determined fairly accurately by performing a critical speed analysis on a numerical model of the machine, or through direct experimental measurements. These resonance frequencies can then be added to the expert system as facts, and a set of rules can automatically determine if resonance is a problem. In this case, coupling a rulebased system with numerical analysis programs enables the use of both a heuristic search strategy and the deeper knowledge gained through numerical analyses of the machinery. The deeper knowledge may also provide insight into the development of new rules.

The third data option, online information is a step toward the development of an online, realtime, machine monitoring and diagnostic system. The online information option provides an interface to the online monitoring system. The online monitoring system uses vibration measurements from the rotating machine and automatically performs vibration analysis. The conclusions reached by this system are passed as facts with a certainty factor to the expert system via the online information interface. Interfacing the online monitoring system with the expert system environment will reduce the number of queries that the user must answer during consultation of the expert system.

The two remaining options in the preprocessor help the expert maintain the knowledge base. These options are a knowledge base consistency checker and a knowledge base editor. The consistency checks option tests the existing knowledge base for a number of inconsistencies which include redundant, contradicting, and circular rules. Consistency checks are a subset of the techniques used for validation and verification of the expert system.

The knowledge base editor enables the expert to perform modifications to the knowledge base without knowing the details of the knowledge representation scheme or programming language.

### *Processor*

The main purpose of the processor is to perform the actual diagnosis. The diagnosis is performed in three steps as shown in Figure 6. In the first step, the user determines which data sources are to be incorporated in the diagnosis. The facts derived from these sources are then loaded into the system. These include the facts derived from analytical models, the maintenance history of the machine and the results of computerized or expert interpretation of the vibration measurements. The machine's components and running conditions are also loaded. Finally, the knowledge base is examined, and the rules which apply to the specific rotating machine are loaded.

In the second step, the processor performs a preliminary diagnosis of the machine fault. This diagnosis determines which of the generic problem classes contain the possible problems in the machine. The generic problem classes are the result of grouping problems that produce similar symptoms. For example, the problems: loss of part, rotor bow, and crack are grouped under the generic problem class of unbalance, because these problems create an unbalance at the rotor. At the running frequency, this unbalance produces above normal vibrations that are in a radial direction and located at the rotor. The processor uses a backward

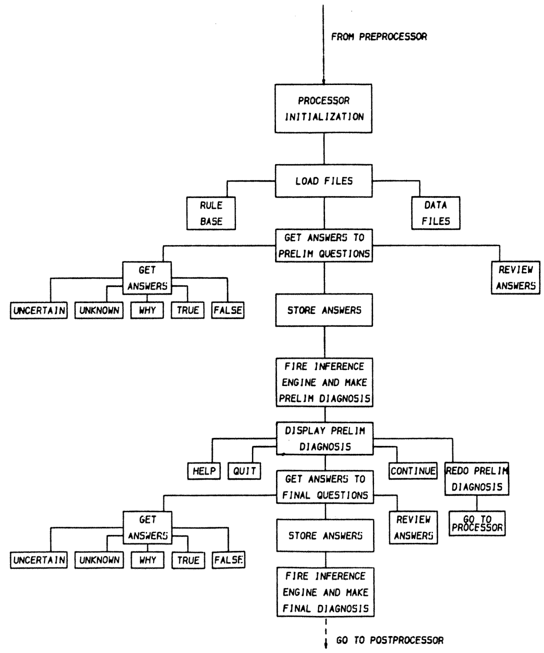


*Figure 6. Processor Diagram.*

chaining inference mechanism with uncertainty handling capabilities to make the diagnosis. After the preliminary diagnosis has been made, the significant generic problem classes are displayed.

The final step in the processor is to refine the results of the preliminary diagnosis, using the same inference mechanism, to arrive at a final diagnosis. One advantage of performing a preliminary diagnosis and determining the significant generic problem classes is that the search can be limited to the problems specific to these classes. In the final diagnosis, the inference mechanism gathers the problem names under the significant generic problem classes and tries to prove each problem. The same backward chaining inference mechanism which made the preliminary diagnosis, makes the final diagnosis. After the final diagnosis has been completed, the results are sent to the postprocessor.

### *Postprocessor*

The postprocessor provides facilities for viewing and verifying the results of the diagnosis (Figure 7). The postprocessor automatically displays a truncated list of the results that have a total belief value which is greater than the threshold value of 0.1. The expert can also view all of the results. This facility aids in validating the expert system, and, in particular, the knowledge base. Explanation facilities are also an essential tool for verification and validation of the expert system. The explanation facility enables the expert to review the expert system's line of reasoning by retracing the expert system's steps through the rulebase that led to the diagnosis.

Another important facility in the postprocessor is the results advisor facility. The results advisor facility provides two functions. First, based on the results of the diagnosis, the results advisor suggests what actions should be taken. For instance, if the machine defect is severe enough, the advisor may suggest that the machine be shutdown. Second, this facility also lists what additional information would increase the certainty of the diagnosis and which tests can be performed to determine that information.

### *General System Usage*

The ROMEX system interface consists of a number of menus and dialogue boxes. There are two types of menus in the pro-

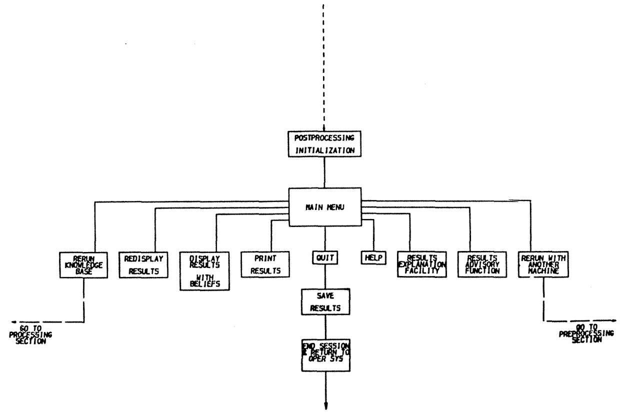


*Figure 7. Postprocessor Diagram.*

gram, radio list boxes and choice list boxes. Radio list boxes allow only one item to be selected from the menu. Choice list boxes allow multiple items to be selected. To select an item in a radio list box, use the arrow keys up and down, to place the arrow > in front of the desired option and press ENTER. To select an item in a choice list box, place the arrow > in front of the item and press the SPACE BAR. The space bar acts like a toggle, selecting and deselecting the item. After all of the desired items have been selected, press ENTER.

Dialogue boxes are a kind of window that contain a number of "controls." The radio list box and choice list box are two types of controls. Other types of controls include edit boxes and push buttons. An edit box creates an edit region, and push buttons are used to perform an action associated with the push button. For example, the word Select in Figure 9 is a push button. If more than one push button is in a dialogue box, a default button is selected. The default is the button with a double line around it. In Figure 8, the Select push button is the default. To move between controls, use Tab or Shift-Tab. A control which contains a blinking cursor is called the current control. The current control is said to be in focus.

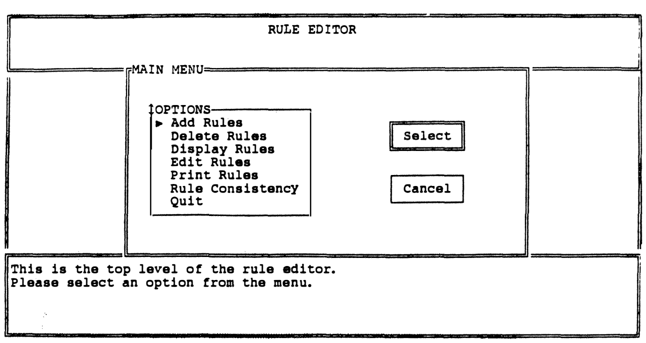


*Figure 8. ROMEX System Main Menu.*

### *Using the Program*

The top level menu for the ROMEX system, shown in Figure 9, is the first menu displayed on the screen. All of the options are available to the user except Option 3, Online Information and Option 4, Build Numerical Model.

*Option 1. Enter Machine Information*

This option allows the user to add a new machine to be diagnosed. Before the expert system can be consulted for a machine, the Enter Machine Information option must be executed. This

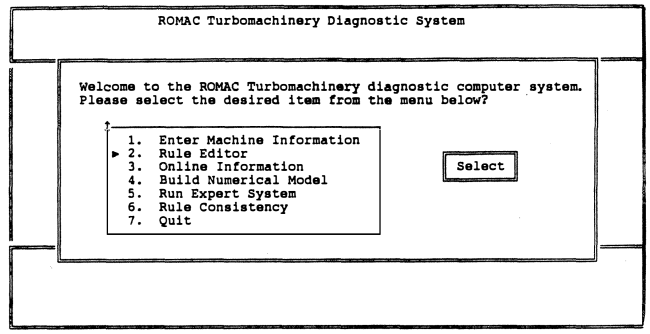


*Figure 9. Rule Editor Main Menu.*

option queries the user to enter the name of the machine. Once the name is entered, the program asks a number of questions about the components of the machine. After all of the questions have been answered, the program returns to the Top Level Menu.

*Option 2. Rule Editor*

The Rule Editor Option enables the user to edit the rules in the knowledge base. This option is best described by working through an example. In the example, the rule "If the spectrum shows multiples of half synchronous vibration, then the problem is partial rub with a belief of 0.5," is added to the rule base. *Note:* this rule is a correct rule for the rulebase that has been removed for the purpose of this tutorial. Therefore, the user should work through this tutorial before consulting the expert systems.

*Step a. Load System.*

Load the ROMEX system using the steps given in Section 1.

*Step b. Select Rule Editor.*

Select the Rule Editor Option in the menu by positioning the arrow > in front of the option and pressing ENTER (Figure 10). This loads the Rule Editor and displays the Main Menu of the rule editor on the screen.

*Step c. Select Add Rules Option (Figure 8).*

Select the Add Rules Option in the menu and press ENTER. The ROMEX system creates a rule by gathering the components of a rule separately, and combining these components internally to form the rule. The components of a rule are: problem name, symptoms, belief, and machine components. These components are entered in Steps d, f, and g, respectively.

*Step d. Enter Problem Name Component*

Type: *partial rub* and press ENTER (Figure 10). To view a list of the problem names in the system, use the TAB key to move the focus of the dialogue box from the ENTER to the LIST push

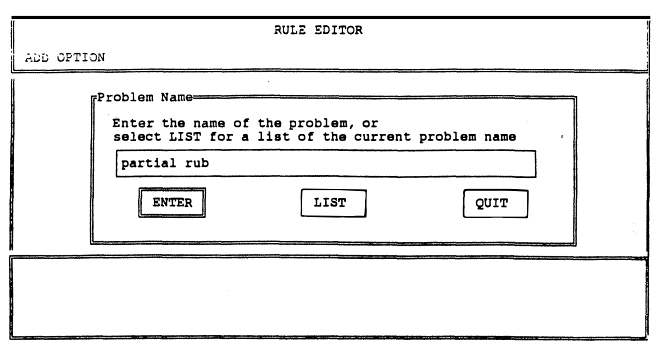


*Figure 10. Problem Name Dialogue Box.*

button. When the LIST push button is in focus, a double line is drawn around the word. Pressing the ENTER key will cause a list of the general problem categories to be displayed.

*Step e. Enter Symptoms Component*

Type the symptom: *the spectrum shows multiples of half synchronous vibration* and press ENTER (Figure 11). The system will accept the symptom and prompt you for another one. Since there is only one symptom in this example, move the focus of the dialogue box to QUIT and press ENTER. After the symptoms have been entered, the program calls the natural language interface. The results of the natural language processing are displayed in the bottom window of the screen (Figure 12). The natural language interface is designed to handle simple active and passive sentences of the form >subject>>verb>>object>. Examples of the types of sentences that the interface can analyze are shown in Figure 13.

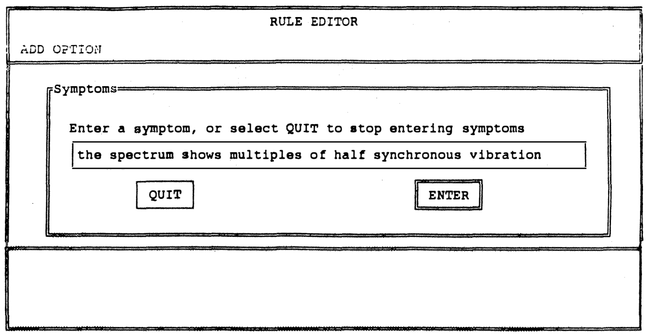


*Figure 11. Symptoms Dialogue Box.*

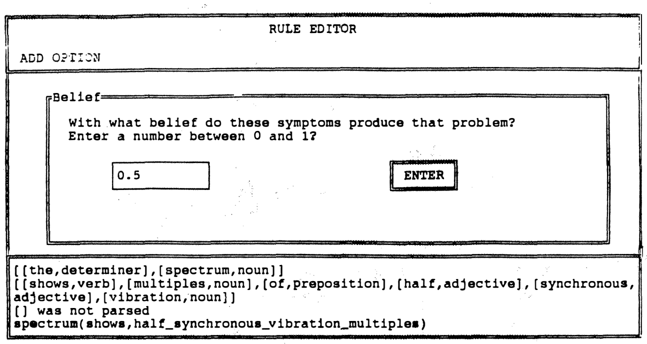


*Figure 12. Belief Value Dialogue Box.*

1) The predominant frequency of vibration is 3x.
2) The direction of the vibration is radial.
3) The amplitude of vibration increases as the flowrate increases.
4) The phase difference between the horizontal and vertical vibration is 90 degrees.
5) The radial vibrations at the bearings are in–phase.

*Figure 13. Legal Sentence Examples.*

*Step f. Enter the Belief Value Component*

Type: *0.5* and press ENTER. The belief value is a number between 0.0 and 1.0 inclusive. For the ROMEX system, the belief value should be greater than 0.1 (the threshold value).

*Step g. Select the Machine Components*

Select general from the list of components and press ENTER (Figure 14). Selecting general enables the rule to be used in every consultation of the expert system, regardless of the components in the machine. The machine component part of a rule allows the user to specify which classes of machines that the rule applies. Since this menu is a choice list box, more than one component may be selected from the menu.

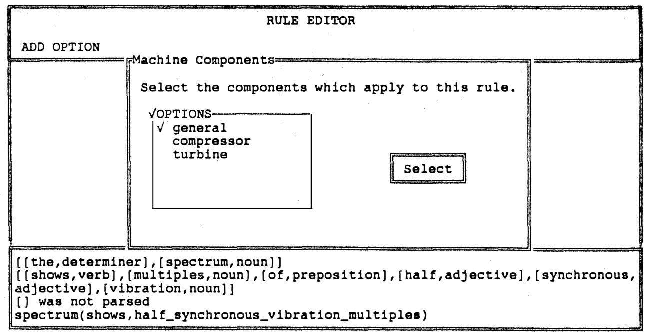


*Figure 14. Machine Component Dialogue Box.*

*Step h. Rule Verification*

Move the focus of the dialogue box to the Accept push button and press ENTER (Figure 15). This box allows you to accept or reject the rule that you have entered. After the rule has been accepted, the program adds the rule to the database.

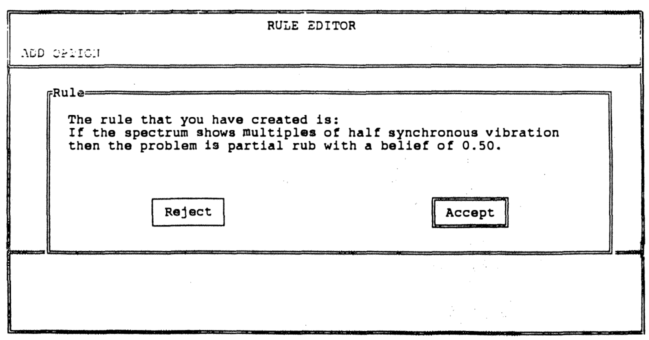


*Figure 15. Rule Verification Dialogue Box.*

*Step i. Adding a Symptom Question*

Select the Yes push button and press ENTER. If a new symptom is added to the system, the program will also create a question for that symptom as shown in Figure 16. After this step, the prolog version of the rule is displayed and the program returns to the Main Menu of the Rule Editor (Figure 17).

*Step j. Exiting the Rule Editor*

Use the arrow keys to select Quit from the menu and press ENTER. The program returns to the Top Level Menu.

*Option 5. Run Expert System*

This option calls the actual expert system. This option is explained using an example for a machine named Kemiral running at 6000 rpm. This machine shows a predominant vibration at 0.5× (3000 rpm) and vibrations at multiples of 0.5× (i.e., .5×, 1×, 1.5×, ..).

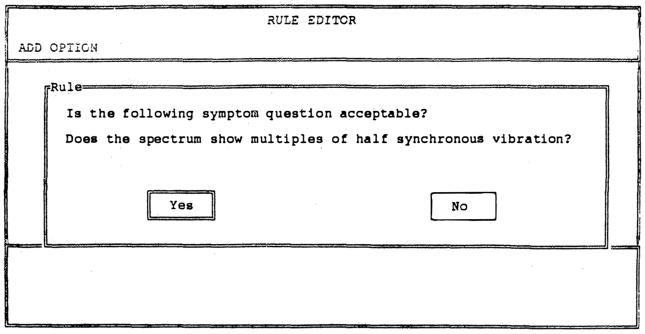


*Figure 16. Question Display.*

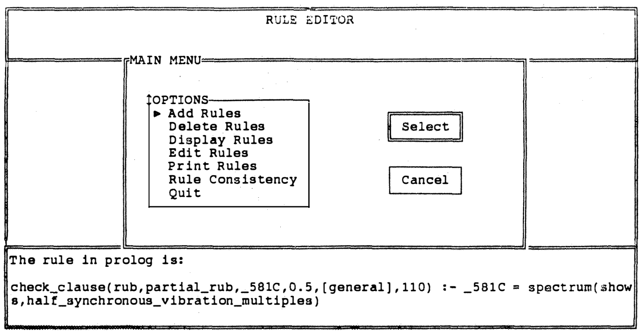


*Figure 17. Rule Editor Main Menu with Prolog Rule.*

*Step a. Select Run Expert System.*

Select the RUN Expert System option in the menu by positioning the arrow > in front of the option and pressing ENTER (Figure 18). This loads the Expert System.

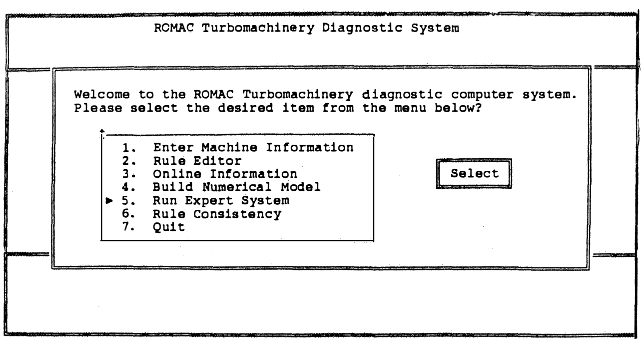


*Figure 18. Main Menu with Expert System Selection.*

*Step b. Select Machine Name*

Position the arrow in front of the name "Kemira1" and press ENTER (Figure 19). The system loads the information for this machine. If a new machine is to be diagnosed, focus the dialogue box on the Machine Name edit box, type in the name of the new machine, and press ENTER. If the machine is unknown, the system will run the Machine Information Option to gain the necessary background information.

*Step c. View Machine Information*

Select the Yes push button and press ENTER (Figure 20). The system will display information about Kemira1. To continue with the diagnosis press ENTER (Figure 21).

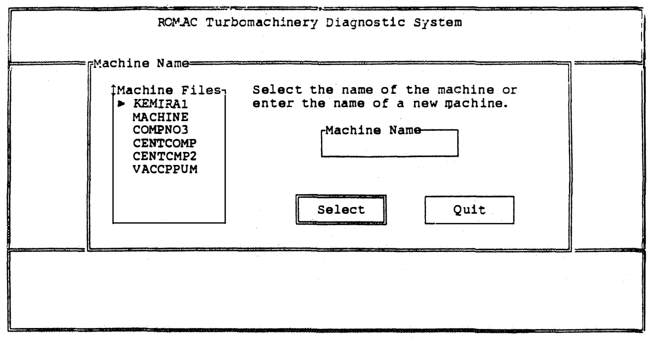


*Figure 19. Machine Name Dialogue Box.*

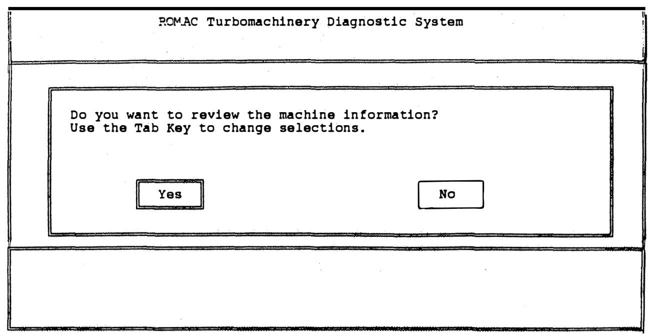


*Figure 20. Review Machine Information Dialogue Box.*

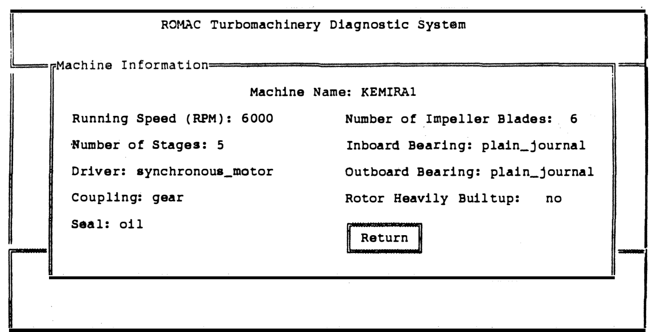


*Figure 21. Machine Information Display.*

*Step d. Enter Vibration Information*

Enter the data shown in Figure 22. Use the Tab key to move between the controls. Select the ENTER push button and press the ENTER key. The vibration data entered is the vibration fre-

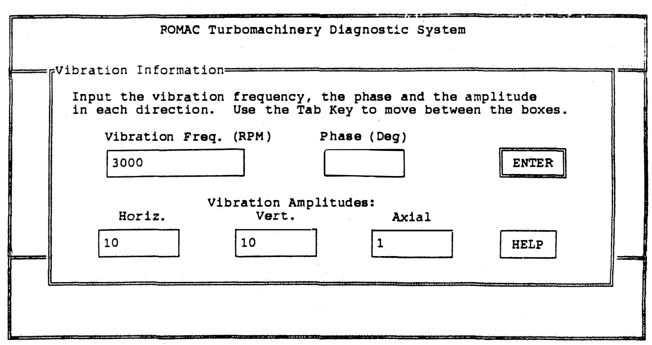


*Figure 22. Vibration Information Dialogue Box.*

quency, phase, and vibration amplitudes in the horizontal, vertical and axial directions.

*Step e. More Vibration Data*

Select NO and Press ENTER (Figure 23). For this example, only one frequency is entered into the system. *Note:* When performing a diagnosis, you may want to run the diagnosis a couple of times varying the amount of vibration data to look for sensitivity in the results.

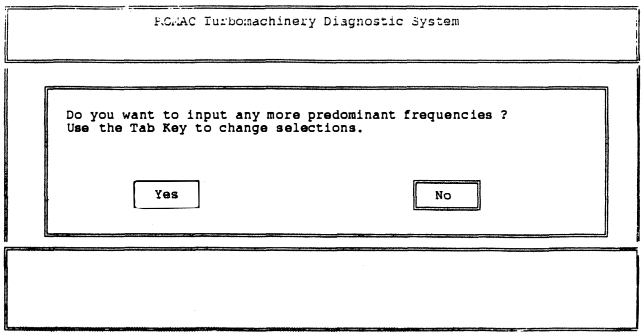


*Figure 23. Additional Vibration Frequencies Dialogue Box.*

*Step f. System Queries.*

Answer No to the question "Is the predominant frequency random?" Press Enter (Figure 24). After this question is answered, the system asks a number of questions. Answer NO to each of these questions until a preliminary diagnosis is made. The other responses that can be made allow the user to answer "Unknown" or "Uncertain". Unknown means that the user does not know what the answer is. If unknown is entered, the program treats the rule like it does not exist. Uncertain is used when the user has an idea of what the answer is, but is not completely certain. If uncertain is entered, the program asks the user if he/she thinks that the rule is true or false and the certainty of the answer. The user can also select WHY to view the rule that the system is currently trying to prove.

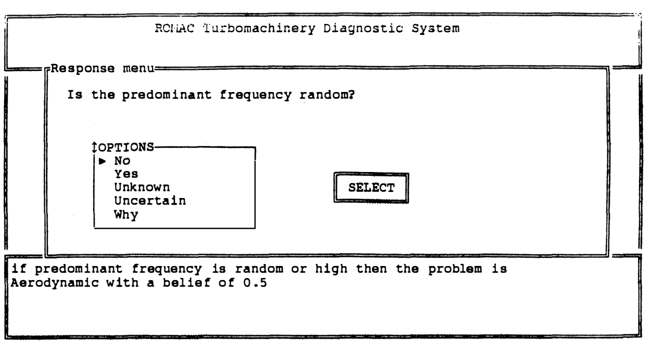


*Figure 24. Response Menu Dialogue Box.*

Step g. Preliminary Diagnosis

After all of the questions have been answered, the results of the preliminary diagnosis are displayed on the screen (Figure 25).

Next, the program again asks a series of questions. The response to all but one of these questions is NO. The response to the question "Does the frequency spectrum show multiples of half synchronous vibration?" is Yes (Figure 26). *Note:* this is the

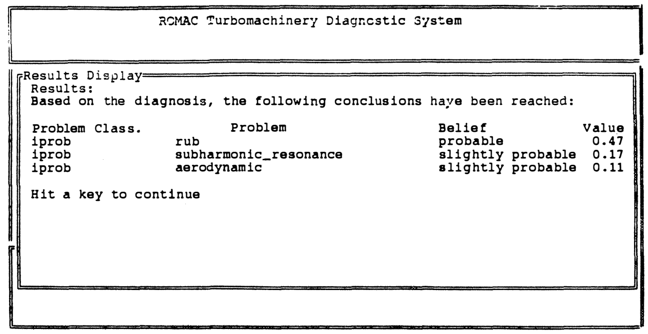


*Figure 25. Preliminary Results Display.*

same symptom that was added to the system using the Rule Editor Option.

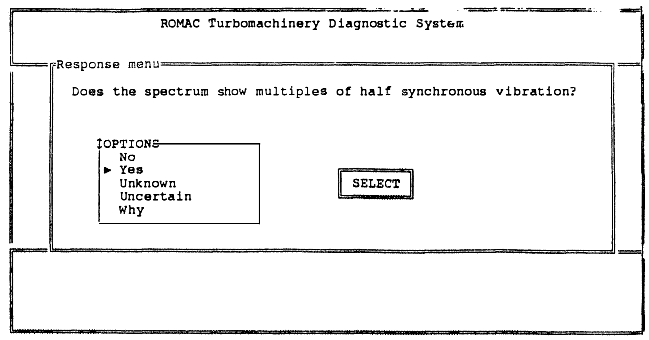


*Figure 26. Response Menu Dialogue Box Showing New Rule.*

*Step h. Final Diagnosis*

Once all of the questions have been answered, the system arrives at a final diagnosis (Figure 27). Typing any key causes the program to display the Post Processor dialogue box (Figure 28).

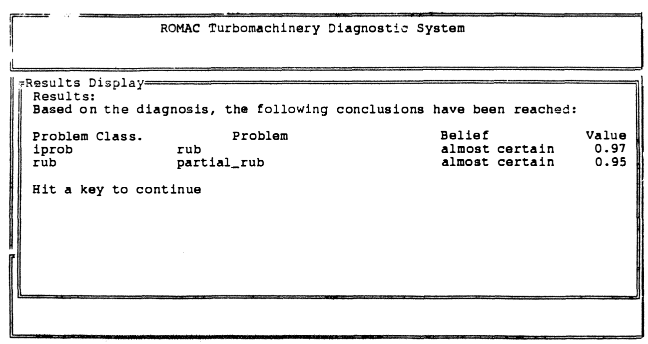


*Figure 27. Final Diagnosis Display.*

*Step i. Explanations of the Results*

Select the Explanations push button and press ENTER.

*Step j. Explanation of a Partial Rub.*

Move the arrow in front of the problem name partial rub and press ENTER (Figure 29). This causes the system to display one of the partial rub rules which was "fired" during the diagnosis to be displayed (Figure 30). Selecting Yes again displays the rules

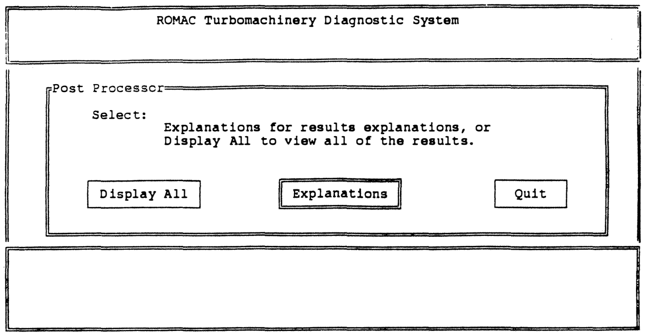


*Figure 28. Postprocessor Main Menu.*

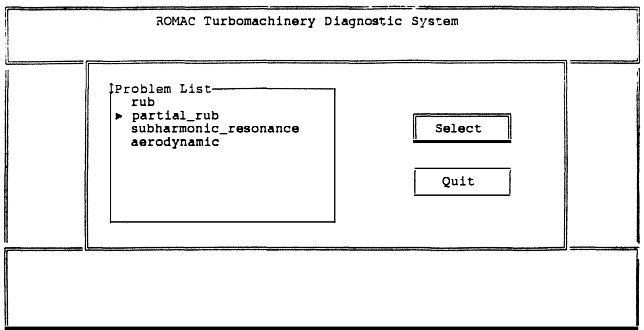


*Figure 29. Explanation Facility Main Menu.*

that was just added to the system (Figure 31). Selecting No returns the program to the Post processor dialogue box.

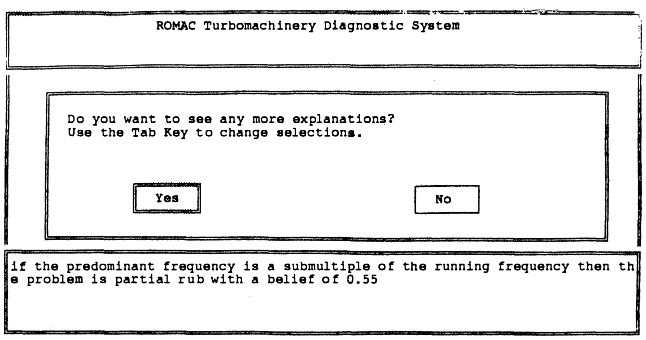


*Figure 30. Explanation Facility Example.*

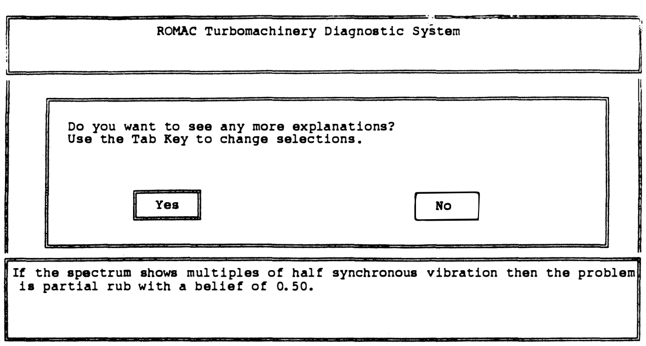


*Figure 31. Further Explanations Dialogue Box.*

*Step k. Exit the Expert System.*

Select the QUIT push button. The program returns to the top level menu of the ROMEX system.

*Option 6. Consistency Checker Option*

The consistency checker looks for inconsistencies in the knowledge base. The current version checks for detached and redundant rules. Detached rules cannot be fired by the expert system. Redundant rules have the same problem name and symptoms. The function of this option is simple. When the option is called, the knowledge base is loaded, and the program checks for detached rules. The rule numbers of any detached rules are displayed on the screen. The system then checks for any redundant rules and displays their rule number on the screen.

*Option 7. Quit*

The Quit option terminates execution of the ROMEX program and returns the computer to the Arity/prolog prompt ?-.

*Exiting Arity/prolog*

Type *halt.* at the ?-. This terminates Arity/prolog and returns to DOS.

## CONCLUSIONS

The development of a fault diagnostic system for turbomachinery requires a resolution of a number of issues in expert systems, analytical/ experimental methods, data acquisition systems, and user interaction. A two-pronged approach has been established for this development: (i) a research oriented facet of the development explores the various issues and concepts to establish an evolving, comprehensive framework for the diagnostic system, and (ii) a more pragmatic facet which implements the concepts in a working prototype, called ROMEX, such that a testbed for the comprehensive framework research is continually available for "hands on" tests and validation. Industrial participation in evolving the framework and also in testing the prototype are necessarily key ingredients for the successful development of this system.

## REFERENCES

1. Bajpai, A., "An Expert System Model for General-Purpose Diagnostics of Manufacturing Equipment," ASME Manufacturing Review, *1*, (3), pp. 180-187 (October 1988).
2. Duda, R., Hart, P., and Nilsson, N., "Subjective Bayesian Methods in Rule Based Systems," Technical Note 124, SRI (1984).
3. Buchanan, B., and Shortliffe, E., *Rule Based Expert Systems: The MYCIN Experiments*, Reading, Massachusetts: Addison-Wesley (1984).
4. Zadeh, L. A., "The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems," *11*, pp. 199-227 (1983).
5. Shafer, G., *A Mathematical Theory of Evidence*, Princeton, New Jersey: University Press (1976).
6. Gordon, J., and Shortliffe, E., "A Method for Managing Evidential Reasoning in a Hierarchical Hypothesis Space," Proceedings of AAAI-84, Austin, Texas (1984).
7. de Kleer, J., and Brown, J., "Mental Models of Physical Mechanisms and Their Acquisition," *Cognitive Skills and Their Acquisition*, Hillsdale, New Jersey: Erlbaum Associates, pp. 285-309 (1981).
8. Chandrasekaran, B., and Mittal, S., "Deep Versus Compiled Knowledge Approaches to Diagnostic Problem-Solving," Int. J. Man-Machine Studies, Academic Press, Inc., *19*, pp. 425-436 (1983).

9. Suwa, M., Scott, A., and Shortliffe, E., "Completeness and Consistency in a Rule-Based System," *Rule Based Expert Systems*, Reading, Massachusetts: Addison-Wesley Publishing Co., pp. 159-170 (1984).
10. Nguyen, T., Perkins, W., Laffey, T., and Pecora, D., "Checking an Expert Systems Knowledge Base for Consistency and Completeness," Proceedings of the Ninth International Joint Conference on Artificial Intelligence, Los Angeles, California, pp. 375-378 (1985).
11. Cragun, B., and Steudel, H., "A Decision-Table-Based Processor for Checking Completeness and Consistency In Rule-Based Expert Systems," Int. J. Man-Machine Studies, Academic Press, Inc., *26*, pp. 633-648 (1987).
11. Buffer, M., "Turbomachinery Diagnostic Expert System with User Friendly Knowledge Base Editing Facility," M.S. Thesis, Mechanical and Aerospace Engineering, University of Virginia (August 1989).
13. Schutte, P., Abbott, K., Palmer, M., and Ricks, W., "An Evaluation of a Real-Time Fault Diagnosis Expert System for Aircraft Applications," 26th IEEE Conference on Decision and Control, Los Angeles, California (1987).
14. Nahar, R., "An Expert System For Vibration Fault Diagnosis of Turbomachinery," M.S. Thesis, Mechanical & Aerospace Engineering, University of Virginia (1989).
15. Koch, C., Isle, B., and Butler, A., "Intelligent User Interface for Expert Systems Applied to Gas Turbine Maintenance and Troubleshooting," Proceedings of the EPRI Conference on Expert Systems In Power Plants, Boston, Massachusetts (1987).
16. Shah, R., Pomeroy, B., Salemme, R., and Irving, R., "EXACT - An Expert System For Gas Turbine Troubleshooting," Proceedings of the EPRI Conference on Expert Systems In Power Plants, Boston, Massachusetts (1987).
17. Sterling, L., and Beer, R., "Incremental Flavor-Mixing of Meta- Interpreters for Expert System Construction," CH2340, IEEE (1986).


## ACKNOWLEDGEMENTS

The financial support for this work was provided by the ROMAC Laboratory through its industrial partners, and by the Mechanical and Aerospace Engineering Department of the University of Virginia. The Center for Computer Aided Engineering provided the computer facilities for this project. Mike Buffer, Rahul Nahar, and Peter Accorti, graduate students who have worked on this project, are acknowledged for their contributions.